

REMARKS

The Examiner is thanked for his careful and very thorough Office Action.

Claims 1-5, 7-13, 16, 18, and 19 have been rejected. By the foregoing amendments, various Claims are sought to be amended or canceled without prejudice.

The Examiner has stated that Claims 14, 15, 17, and 20 would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claim. By the foregoing amendments, Applicant has amended Claims 14, 15, 17, and 20 as suggested by the Examiner, and their allowance is respectfully requested.

The foregoing amendments to the specification are submitted to improve clarity, and to remove various typographical and other minor informalities. These changes are respectfully asserted not to introduce new matter, and their entry is respectfully requested.

§112 Rejections

The Examiner rejected Claim 11 under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The Examiner has suggested that the user input device, the microprocessor, and data output device claimed in Claim 11 are not shown in the drawings.

Applicant would like to direct the Examiner's attention to Fig. 1 and the accompanying description in the present application. A user input device would be, for example, keyboard 435 and mouse 440. The microprocessor would be microprocessor 425. A data output device would be, for example, display 450 and speaker 477. Therefore, Applicant believes that the subject matter in Claim 11 is shown in Fig. 1 and described in the accompanying description.

The Examiner has rejected Claims 2 and 8-10 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. With regard to Claim 2, the Examiner has suggested that it is unclear how the

Amendment – Serial No. 09/591,225.....Page 9

graphics accelerator unit manages page faulting of texture data from dedicated graphics memory into a main memory.

The present application describes how this task is accomplished beginning on page 8, line 11 as follows:

When a logical page fault occurs and the page of texture is in the second level of memory (i.e. the host's physical memory) it will be fetched in automatically by the graphics memory manager, and the host is not aware anything has happened. In a preferred embodiment, a number of automatic mechanisms would be in place for this to happen:

- a. Determine where the page is located in host physical memory.
- b. Determine which page out of the working set (in level 1 memory) to use. In a sample embodiment, this determination uses the least recently used algorithm.
- c. Make this page the most recently used page (as well as continuing to keep the least-recently-used list up to date as other pages are used).
- d. Update the page tables for the new page and remove any reference to the page just bumped out of memory (if any).
- e. Download the page.
- f. Restart texture processing.

Note that if the faulting logical page identifies a page in the third level memory the host does (a) (after having made the page available), but the hardware carries on and does b, c, d, e and f.

Of course, these points in the specification are intended as examples only, and are not intended to limit the scope of the claims.

Applicant respectfully submits that it is not the role of the claims to teach one skilled in the art to reproduce the invention, but rather to define the legal metes and bounds of the invention. *In re Rainer*, 305 F.2d 505, 509, 134 U.S.P.Q. 343, 346 (C.C.P.A. 1962). If the metes and bounds of the claimed invention are clearly ascertainable, then the claim cannot be properly rejected as "vague" or "indefinite" under 35 U.S.C. § 112, second paragraph. Accordingly, Applicant believes that Claim 2 and its dependents are not indefinite under 35 U.S.C. § 112, second paragraph.

Art Rejections

The art rejections are all respectfully traversed.

Rejection Under 35 USC 102(b)

Claims 1, 3, and 13 stand rejected under 35 USC Section 102(b) as anticipated by *Mizuyabu et al.*

The claim language of Claim 1 is not taught or suggested by *Mizuyabu et al.* Specifically, Claim 1 recites, "a graphics accelerator unit which manages page faulting of texture data invisibly to the host processor."

Mizuyabu et al. (U.S. Patent No. 6,297,832) relates to sequencing memory accesses in a video graphics system. *Mizuyabu et al.* does not appear to disclose any innovations with regard to graphics accelerators.

A graphics accelerator is generally considered, for example, as a type of video adapter that contains its own processor to boost performance levels. These processors are specialized for computing graphical transformations, so they achieve better results than the general-purpose CPU used by the computer. In addition, they free up the computer's CPU to execute other commands while the graphics accelerator is handling graphics computations.¹ The paragraph beginning on page 2, line 18 of the present application states, "Since rendering is a computationally intensive operation, numerous designs have offloaded it from the main CPU."

¹ This particular teaching is illustrated at http://www.webopedia.com/TERM/G/graphics_accelerator.html.

Mizuyabu et al. do not appear to offload the functions of their video graphics system from the main CPU. This is perhaps due to the fact that *Mizuyabu et al.* are concerned only with minimizing page faults and not with offloading work from the main CPU:

...memory controller 20 receives requests for memory access from the plurality of clients and organizes these requests in such a way as to minimize page faults. (col. 5, ll. 1-3)

By contrast, the present inventions are not only concerned with managing page faults, but also the importance of offloading work from the main CPU by performing the page managing operations invisibly to the host via the graphics driver. As stated in the paragraphs beginning on page 7, line 11 of the present application, this is not a trivial task:

The present inventor has realized that managing the texture memory in the driver or by the application is very difficult (or impossible) to do properly, because:

1. What does the driver/application do when it runs out of memory and needs to fit another texture in? What texture(s) does it delete?
2. The texture has to be completely resident and physically contiguous so a large enough space must be available.
3. A texture which is about to be used **MUST NOT** be deleted or moved: otherwise all command buffers will be outdated.
4. In some cases a texture map will not fit into memory even when all other textures are deleted (a 2Kx2K 32bpp texture map takes 16Mbytes of memory).
5. The texture heap must be compacted to reclaim storage.

The idea of applying virtual management techniques to textures in 3D graphics hardware appears to be suggested, for example, by U.S. Patent 5,790,130 to Gannett. ...

However, the present inventor has realized that these schemes are ill-suited for most personal computer applications (and many workstation applications). The main aim in these implementation seems to have been to allow very large texture maps (16Mx16M or larger) to be used. By contrast, the innovations in the present application are not motivated only by desire for such large maps, but to remove the software problems in managing the comparatively small amount of texture storage (vs the large amounts of texture storage in SGI and HP machines) efficiently. Thus it is possible that the architectural innovations disclosed herein can be used in combination with those used by SGI and HP.

Accordingly, the present innovations solve problems that are not even considered by the video graphics system of *Mizuyabu et al.*

Also, the Examiner has suggested that *Mizuyabu et al.* memory controller 20 manages page faulting invisibly to the host processor because the host processor is not shown in Fig. 1. However, Applicant respectfully submits that this conclusion is without basis.

Fig. 1 shows a video graphics system which must be coupled to a host processor. As established earlier, offloading work from the main CPU to the graphics driver is innovative not conventional. Therefore, without any teaching in *Mizuyabu et al.* to the contrary, the reasonable conclusion would be that the operations of video graphics system disclosed in *Mizuyabu et al.* are not performed invisibly to the host processor.

A prior art reference anticipates the claimed invention under 35 U.S.C. §102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). In the present case, the Examiner has not cited teaching in the reference that discloses the limitations of Claim 1. Therefore, Applicant respectfully submits that Claim 1 is not anticipated by *Mizuyabu et al.*

Claim 3 also recites features not shown or suggested by *Mizuyabu et al.* Specifically, Claim 3 recites, "a graphics accelerator unit, comprising rendering accelerator logic, dedicated graphics memory, and a second memory management unit which manages texture data for said accelerator logic and performs page faulting of said texture data, invisibly to said CPU."

Again, *Mizuyabu et al.* does not appear to disclose a graphics accelerator unit. Because *Mizuyabu et al.* does not appear to disclose a graphics accelerator unit, processors 70 and 80 could not be considered to be rendering accelerator logic. Also, memory controller 20 could not be said to manage texture data for the accelerator logic when there is no accelerator logic present. Finally, as stated earlier, the operations of memory controller 20 could not be determined to be invisible to the CPU.

Therefore, for the reasons stated above, the Examiner has not cited teaching in the reference that discloses the limitations of Claim 3. Accordingly, Applicant respectfully submits that Claim 3 is also not anticipated by *Mizuyabu et al.*

Finally, dependent Claim 13 depends directly from independent Claim 3 and incorporates all the limitations thereof. Therefore, for the reasons stated above, Applicant respectfully submits that Claim 13 is also not anticipated by *Mizuyabu et al.*

Thus, for the reasons discussed above, Applicant respectfully requests withdrawal of this rejection.

Rejections Under 35 USC 103(a)

Claim 12 stands rejected under 35 USC Section 103(a) as being unpatentable over *Mizuyabu et al.* in view of *Porterfield*.

Porterfield (U.S. Patent No. 6,249,853) relates to a modular device for storing, addressing, and retrieving graphics data from main memory. *Porterfield* does not appear to disclose any innovations with regard to graphics accelerators.

Dependent Claim 12 depends directly from independent Claim 3 and incorporates all the limitations thereof. Accordingly, even if one were to combine the teachings of *Porterfield* with the video graphics system of

Mizuyabu et al., it still would not result in the present inventions as *Mizuyabu et al.* does not appear to disclose the graphics accelerators of the present inventions. Therefore, a prima facie case of obviousness has not been established by the Examiner with regard to Claim 12.

All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Therefore, Applicant respectfully submits that Claim 12 is not obvious in view of *Mizuyabu et al.* and *Porterfield*, and respectfully requests withdrawal of this rejection.

Claims 4, 5, 7, 16, 18, and 19 stand rejected under 35 USC Section 103(a) as being unpatentable over *Peddada et al.* in view of *Emberling et al.*

Peddada et al. (U.S. Patent No. 6,295,068) relates to a 3D graphics driver that manages a texture cache in the local graphics memory. The driver disclosed by *Peddada et al.* does not appear to manage texture memory in the main memory nor does it suggest managing page faulting of texture data. *Peddada et al.* also does not disclose allowing the graphics accelerator itself direct access to the main memory or any other innovations with regard to graphics accelerators.

Emberling et al. (U.S. Patent No. 6,246,422) relates to a method for storing mip map series in a multi-bank texture memory. *Emberling et al.* does not appear to disclose any innovations with regard to graphics accelerators.

The asserted combination of references does not support each limitation of Claim 4. Specifically, Claim 4 recites, "a graphics accelerator unit having respective local memory associated therewith, and also having a graphics memory manager; wherein, when said graphics accelerator unit attempts to access texture data which is in said physical memory associated with said host, said graphics memory manager fetches said texture data automatically."

The Examiner appears to suggest that storage logic 64 of *Emberling et al.* is a memory manager. However, Applicant respectfully disagrees with this suggestion. Storage logic 64 appears to do nothing more than to split large mip maps into smaller portions and to store various portions of mip maps into predefined memory banks. This understanding is supported by the claim

language used to describe the function of storage logic 64. For example, claim 14 of *Emberling et al.* states:

wherein said storage logic is configured to receive a series of mip maps from said main system memory, wherein successive mip maps of said series have successively smaller sizes, wherein the series of mip maps include large mip maps having sizes larger than a page and small mip maps having sizes less than or equal to a page, wherein the storage logic is configured to (a) split each of said large mip maps into a first portion and a second portion, (b) store the first portions of any two consecutive large mip maps into distinct memory banks of said texture memory, (c) store the second portions of any two consecutive large mip maps into distinct memory banks of said texture memory, (d) store small mip maps, after a first small mip map in said series, into a first memory bank of said texture memory.

Similar descriptions of storage logic 64 can be found in claims 19, 22, and 27 of *Emberling et al.*

"Memory manager" is a term of art used, for example, to describe a part of a computer program that accepts requests from the program to allocate and deallocate chunks of memory. The objectives of the memory manager are generally to allow dynamic memory allocation.² Storage logic 64 does not appear to perform such functions. Therefore, it could not properly be considered to be a memory manager.

Hence, even if one were to combine the teachings of *Peddada et al.* with the teachings of *Emberling et al.*, it still would not result in the present inventions as *Emberling et al.* does not appear to disclose a graphics memory manager. Therefore, a prima facie case of obviousness has not been established by the Examiner with regard to Claim 4. Accordingly, for the reasons stated above, Applicant respectfully submits that Claim 4 is not obvious in view of *Peddada et al.* and *Emberling et al.*

² This particular teaching is illustrated at <http://encyclopedia.thefreedictionary.com/Memory%20manager>.

The asserted combination of references also does not support each limitation of Claim 7. Specifically, Claim 7 recites, "a graphics accelerator unit having respective physical memory associated therewith, and also having virtual memory management; and wherein, when said graphics accelerator unit attempts to access texture data which is in said host physical memory, if said texture data is in said host physical memory, said graphics memory manager fetches said texture data therefrom automatically; and if said texture data is not in said host physical memory, said texture data is first loaded into said host physical memory, and thereafter said graphics memory manager fetches said texture data automatically from said host physical memory."

Again, the Examiner appears to suggest that storage logic 64 of *Emberling et al.* is a memory manager. For the reasons stated above, Applicants respectfully disagree with this suggestion. Also, the graphics accelerator mentioned in *Peddada et al.* does not appear to have virtual memory management capabilities.

Virtual memory, for example, uses disk storage space to make the computer work as if it had more memory. When a file or program is too big for the computer to work with in its memory, part of the data is stored on disk. This virtual storage is divided into segments called pages; each page is correlated with a location in physical memory, or RAM. When an address is referenced, the page is swapped into memory; it is sent back to disk when other pages must be called. This allows the program to run as if all the data is in memory.³

Neither of the applied references, or any motivated combination thereof, discloses or suggests a graphics accelerator unit having virtual memory management with a graphics memory manager capable of automatically fetching texture data from the host's physical memory. Therefore, a prima facie case of obviousness has not been established by the Examiner with regard to Claim 7. Accordingly, Applicant respectfully submits that Claim 7 is not obvious in view of *Peddada et al.* and *Emberling et al.*

³ This particular teaching is illustrated at <http://www.computeruser.com/resources/dictionary/definition.html?lookup=5891>.

Finally, dependent Claims 5, 16, 18, and 19 depend directly from independent Claims 4 and 7 and incorporate all the limitations thereof. Therefore, for the reasons stated above, Applicant respectfully submits that Claims 5, 16, 18, and 19 are also not obvious in view of *Peddada et al.* and *Emberling et al.*

Thus, for the reasons discussed above, Applicant respectfully requests withdrawal of this rejection.

Conclusion

Thus, all grounds of rejection and/or objection are traversed or accommodated, and favorable reconsideration and allowance are respectfully requested. The Examiner is requested to telephone the undersigned attorney or Robert Groover for an interview to resolve any remaining issues.

Respectfully submitted,



N. Elizabeth Pham, Reg.No. 49,042

Customer Number 29106

Attorney for Applicant

Groover & Holmes
One Galleria Tower, Suite 1370
13355 Noel Road
Dallas TX 75240
972-980-5840, FAX -5841

19 AUGUST 2004

Amendment – Serial No. 09/591,225.....Page 18